

Java 跨平台特性的实现原理

董正言

(武汉轻工大学数学与计算机学院 湖北武汉 430023)

摘要:跨平台是Java语言的重要特性,和普通高级语言源程序跨平台不同,Java语言的跨平台是指目标程序跨平台。这种跨平台特性使Java语言成为了目前使用最为广泛的Internet应用程序语言。本文介绍Java语言和其他高级语言在跨平台特性上的区别,以及Java语言实现跨平台特性的原理。

关键词:Java 跨平台 Internet编程

中图分类号:TP311

文献标识码:A

文章编号:1672-3791(2014)06(c)-0020-02

Java能够成为目前使用最广泛的一门Web程序设计语言,具备优秀的跨平台特性是一个重要的原因。本人在大学本科的软件教学工作中发现,很多具备一定程序设计基础的本科学生对Java语言跨平台特性的理解还比较模糊,对Java和普通高级编程语言跨平台特性的区别的认识也不十分深刻。本文介绍Java和普通高级语言跨平台特性的区别,阐述Java跨平台特性的实现原理。

1 Java的诞生、发展和特性

20世纪中后期,计算机网络技术逐步成熟,并被应用到科学研究、航天和军工等各个领域,在此基础上出现了Internet(国际互联网)。Internet把世界各地的不同结构的

计算机网络、主机和个人计算机通过通信设施和通信协议互联起来构成国际互联网。20世纪90年代初,在Internet的基础上,又出现了WWW网络(万维网)。WWW网是Internet提供的一种信息服务,它为Internet用户提供了统一的接口以访问网络上的各种信息和资源。WWW网用统一资源定位技术(URL)实现全球信息的精确定位,用新的应用层协议(HTTP)实现分布式信息共享,用超文本技术(HTML)实现信息之间的链接。这些技术使WWW网用户能方便地访问到Internet上的共享信息。

随着WWW应用的快速增长,SUN公司的James Gosling等人意识到需要一种面向WWW网的程序设计语言,于是他们把一种名叫Oak的编程语言加以改造并用于

Web编程(WWW网络编程),并与1995年5月23日正式更名为Java语言。Java语言一经推出,就在Web编程领域取得了巨大的成功。此后的近20年,Java语言蓬勃发展,并始终在Web编程领域占据主导地位。期间,Java语言的版本也从JDK1.0发展到目前的JDK1.7。

Java是一种跨平台的、分布式的、面向对象的编程语言。Java语言本身的很多设计灵感起源于另一种应用最为广泛并且“历史悠久”的编程语言——C++,所以,Java语言的语法格式和程序结构与C++语言都非常相似,它借鉴了C++语言短小精悍、格式灵活的特点。但在语言的应用领域和某些关键特性的侧重点上,Java和C++又具有本质的不同。Java抛弃了C++语言中的一些容

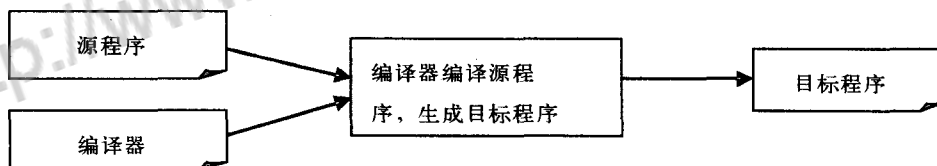


图1 高级语言的编译过程

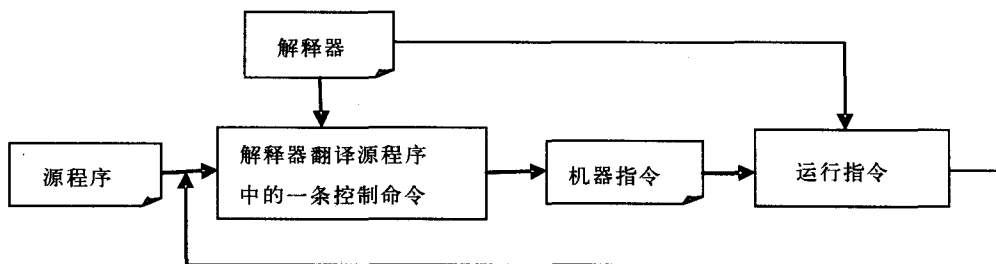


图2 高级语言的解释执行过程

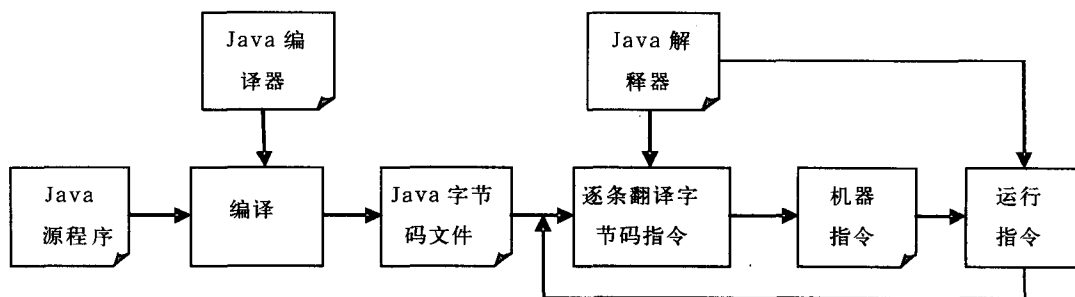


图3 Java程序的编译、解释和运行过程

易引发程序错误的因素。例如Java语言中没有“指针”数据类型,只允许类的单继承,而不允许多继承等;和C++相比,Java更加简洁。和C++的另一区别是:Java是纯粹的面向对象的编程语言,它不支持过程化的编程技术,所以程序中也不包含任何过程化的元素,Java具有面向对象编程技术抽象程度更高、可重用模块粒度更大等优势。此外Java还具有分布式、多线程、解释执行和跨平台等优秀特性。目前的Internet上集成了多种异构平台,作为一门面向Internet应用的编程语言,是否具备良好的跨平台特性就显得尤为重要。本文下面将重点讨论Java的跨平台特性。

2 Java的跨平台特性

自上世纪中叶出现了电子数字计算机后,程序设计语言的发展也经历了几个不同的阶段。最早的时候程序员都是直接使用计算机指令编程,指令是可以被计算机硬件设备识别的二进制编码,计算机程序本质上就是由各种指令构成的一个集合。由于直接使用二进制指令编程难度太大,所以于20世纪50年代出现了汇编语言,汇编语言用一些具有含义的助记符代替了指令中的二进制编码形成汇编指令,程序员使用汇编指令编程的难度大大降低,编程效率得到提升。由于计算机指令和汇编语言都可以直接操作计算机的硬件,所以它们被统称为低级语言。使用低级语言编程的优点是程序指令可以直接操作计算机硬件设备,程序员可以针对特定硬件结构的计算机对程序进行优化以提高执行效率。但是用低级语言编程的缺点更为明显,其一要求程序员必须能够熟练运用计算机指令系统中的各种不同的指令,程序员编程时不光要关注程序的核心逻辑,同时要求程序员掌握特定计算机的硬件结构,变成难度很高;其次,用低级语言编写的程序在不同的硬件平台上不具备可移植性。

20世纪60年代出现了高级编程语言,高级语言程序是由数量有限、容易理解的控制语句组成,程序员还可以使用具有特定含义的短语为程序中的存储单元命名。高级语言中的控制语句只具有逻辑控制能力,而不能直接操作计算机硬件。程序员编写的由高级语言控制命令组成的程序称为“源程序”。由于计算机硬件只能识别二进制指令,所以高级语言源程序中的语句必须被翻译成机器指令后才能运行,这步翻译工作被称为“编译”或“解释”,是由被称

为“编译器”或“解释器”的特殊程序完成的。一种高级语言针对不同的硬件平台会有不同的“编译器”或“解释器”,它们可以把源程序中的控制语句翻译成不同平台上的机器指令。通常把经编译之后的程序称为“目标程序”。这种机制可以保证一个高级语言源程序可在不同的硬件平台上运行,但在运行之前需要使用特定平台的编译器重新编译。这种跨平台特性是源程序级别的跨平台。

和普通高级语言实现的源程序跨平台不同,Java语言的跨平台是指目标程序跨平台。因为不同平台的硬件结构有可能不同,所以这种跨平台听上去是无法实现的。以下介绍Java目标程序跨平台的实现原理。

3 Java跨平台特性的实现原理

高级语言源程序必须被“编译”或“解释”之后才能运行,编译和解释的过程是有区别的。编译是指由编译器把一个高级语言源程序文件一次性翻译成由机器指令组成的目标程序文件。C++语言就是典型的编译型高级语言。编译过程如图1所示。

解释是指由解释器把源程序文件中的控制语句逐条进行翻译并运行,翻译一条运行一条,在解释过程中并不产生完整的目标程序文件。BASIC语言是典型的解释型高级语言。解释和执行过程如图2所示。

Java语言是解释执行的,但在执行之前还需要编译。Java程序编译器先把Java源程序翻译成一个完整的目标程序,和其他编译型高级语言不同,经编译之后的Java目标程序并不是由特定平台的计算机指令组成的,而是由被称为“字节码”的中间代码指令组成,所以目标程序文件又被称为“Java字节码文件”。字节码文件必须由Java解释器来解释和运行。在运行字节码文件时,Java解释器把其中的字节码指令逐条翻译成特定平台的计算机指令并运行。Java解释器又被称为Java虚拟机。任意的计算机平台只要安装了Java解释器,就可以运行Java字节码文件。利用这种机制,Java实现了目标程序跨平台。Java的跨平台特性也被形容为“一次编译,到处运行”。Java语言的编译和解释运行过程如图3所示。

4 结语

通过以上几节的介绍,我们知道:普通的高级编程语言(例如常用的C++语言)的跨平台特性是指源程序级别的跨平台,而

Java语言的跨平台指的是经编译所生成的目标程序的跨平台。Java编译器编译源程序生成目标程序,组成这个目标程序的并不是特定平台的机器指令,而是被称为“Java字节码”的中间代码指令,这种“Java字节码”必须经过Java解释器的解释后才能被转换成特定平台的机器指令,也就是说,对于不同的平台,只要安装了Java解释器,就能够解释并运行Java字节码文件。正是利用这种机制,Java语言实现了目标文件的跨平台。

参考文献

- [1] 周志明.深入理解Java虚拟机[M].北京,机械工业出版社,2011,6.
- [2] 冀振燕,程虎.Java程序编译技术与Java性能[J].软件学报,2000(2).
- [3] 俊,于瑞涛.Java RMI的原理和实现方法[J].青岛大学学报:工程技术版,1999(3).

word版下载: <http://www.ixueshu.com>

免费论文查重: <http://www.paperyy.com>

3亿免费文献下载: <http://www.ixueshu.com>

超值论文自动降重: http://www.paperyy.com/reduce_repetition

PPT免费模版下载: <http://ppt.ixueshu.com>
